

# **μLadder Standard Library**

French version



# TABLE OF CONTENTS

- I. INTRODUCTION..... 5**
- II. STANDARD LIBRARY FUNCTIONS..... 6**
- 1. Temporisations..... 6**
  - 1.1. TON..... 6
    - 1.1.1. Compatibilité..... 6
    - 1.1.2. Description..... 6
    - 1.1.3. Variables d'entrées/sorties..... 7
    - 1.1.4. Exemple..... 7
  - 1.2. T\_BLINK (minute, secondes, millisecondes)..... 9
    - 1.2.1. Compatibilité..... 9
    - 1.2.2. Description..... 9
    - 1.2.3. Variables d'entrées/sorties..... 10
    - 1.2.4. Exemple..... 11
- 2. Math..... 12**
  - 2.1. Min et Max..... 12
    - 2.1.1. Compatibilité..... 12
    - 2.1.2. Description..... 12
    - 2.1.3. Variables d'entrées/sorties..... 13
  - 2.2. AbsDiff..... 14
    - 2.2.1. Compatibilité..... 14
    - 2.2.2. Description..... 14
    - 2.2.3. Variables d'entrées/sorties..... 15
- 3. Scales..... 16**

- 3.1. AIScale..... 16
  - 3.1.1. Compatibilité..... 16
  - 3.1.2. Description..... 16
  - 3.1.3. Variables d'entrées/sorties..... 17
- 3.2. LinearScale..... 18
  - 3.2.1. Compatibilité..... 18
  - 3.2.2. Description..... 18
  - 3.2.3. Variables d'entrées/sorties..... 19
- 4. Changements de valeur..... 20**
  - 4.1. ChangeDetect\_MW..... 20
    - 4.1.1. Compatibilité..... 20
    - 4.1.2. Description..... 20
    - 4.1.3. Variables d'entrées/sorties..... 20
- 5. FILTERS AND AVERAGING..... 22**
  - 5.1. Moyenne glissante..... 22
    - 5.1.1. Compatibilité..... 22
    - 5.1.2. Description..... 22
    - 5.1.3. Variables d'entrées/sorties..... 23
- 6. Stations météo..... 24**
  - 6.1. WeatherStation..... 24
    - 6.1.1. Compatibilité..... 24
    - 6.1.2. Description..... 24
    - 6.1.3. Variables d'entrées/sorties..... 26
    - 6.1.4. Exemple..... 28

**7. LCD DISPLAY..... 29**

- 7.1. WriteText..... 29
  - 7.1.1. Compatibilité..... 29
  - 7.1.2. Description..... 29
  - 7.1.3. Variables d'entrées/sorties..... 30
  - 7.1.4. Exemple..... 31

# **I. INTRODUCTION**

---

This document provides the description of the functions included in the standard library which can be used for the programming of μLadder V8 software and associated software for the programming of PLCs (Programmable Logic Controllers) from mArm7 family.

Special features of each PLC are described within a separated document.

<b>Document version</b>	26/06/14
-------------------------	----------

©SIREA  
All Rights Reserved

*No part of this document or any of its contents may be reproduced, copied, modified or adapted, without the prior written consent of the author, unless otherwise indicated for stand-alone materials.*

<p><b>©SIREA</b> - 69 Rue de l'Industrie - ZI DE MELOU - 81100 CASTRES-FRANCE Tél. +33 (0)5 63 72 93 92 Fax: + 33 (0)5 63 72 93 19</p>
--

## II. STANDARD LIBRARY FUNCTIONS

### 1. Temporisations

#### 1.1. TON

##### 1.1.1. Compatibilité

Toute version du code système

TON générique : Toute version de µLadder

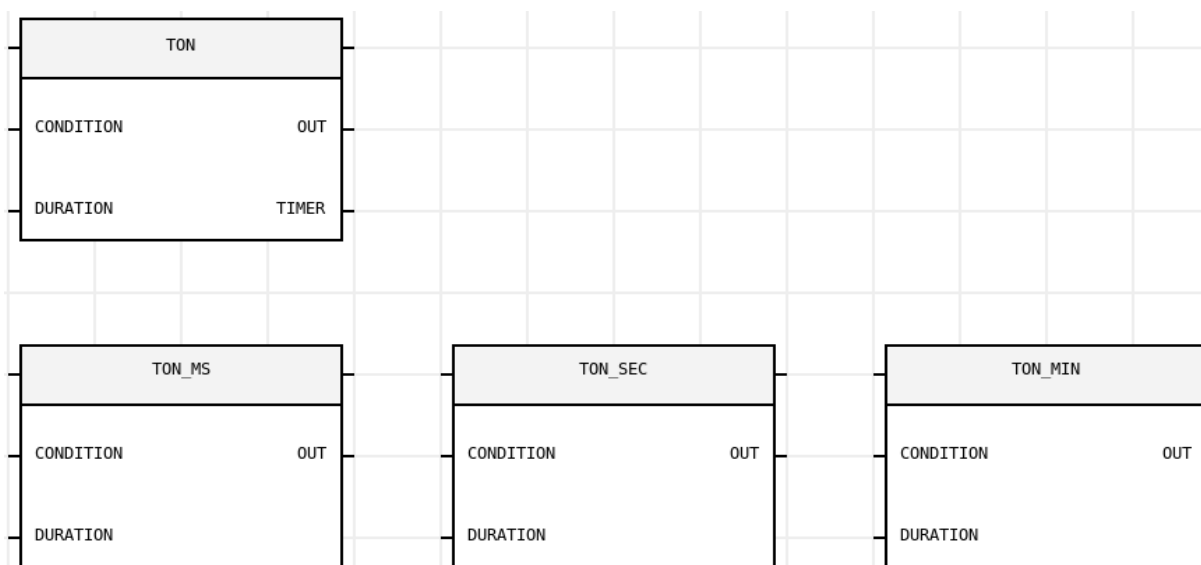
TON base de temps : µLadder V7.9 ou plus

##### 1.1.2. Description

Les temporisation TON permettent de valider une action lorsque une condition est présente pendant un certain temps.

La fonction est déclinée en plusieurs versions :

- Une fonction générique qui accepte toutes les bases de temps mais nécessite de déclarer une variable "timer" dans l'application mère.
- Une fonction par base de temps (millisecondes, seconde, minute) qui intègre déjà la variable "timer"

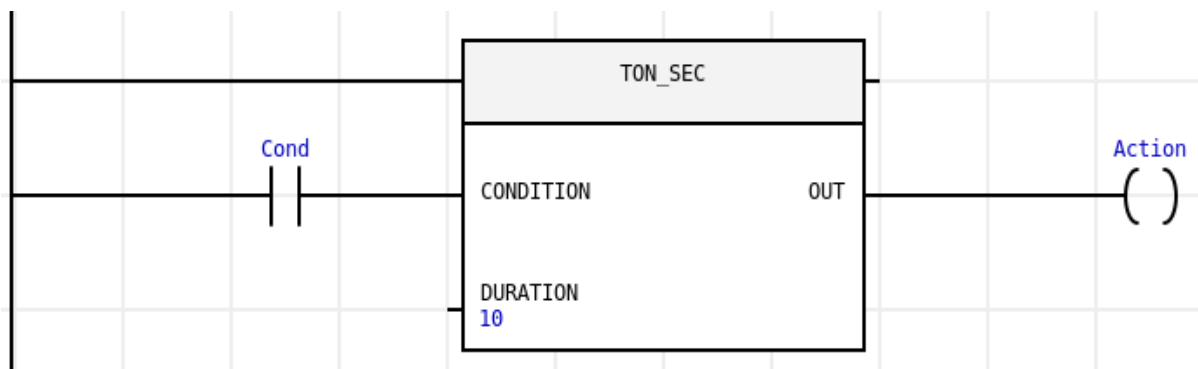


### 1.1.3. Variables d'entrées/sorties

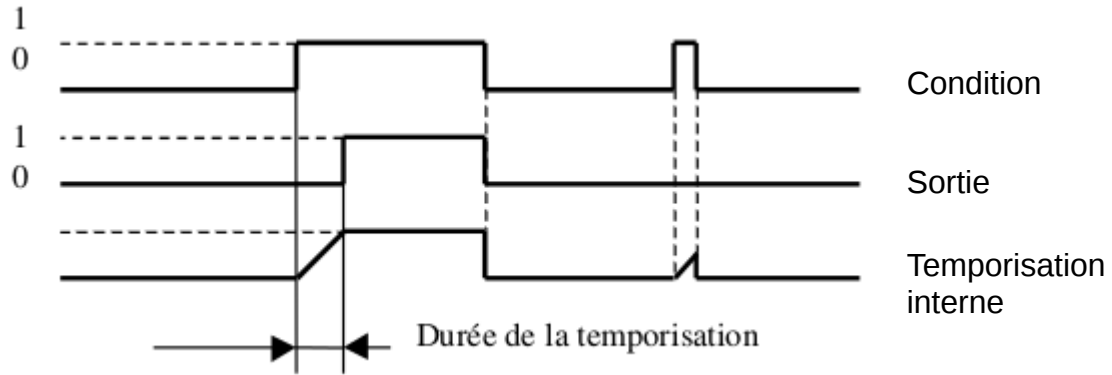
Variables d'entrée		
Nom	Type	Commentaire
<i>CONDITION</i>	M	Condition à temporiser
<i>DURATION</i>	MW	Durée de la temporisation.  L'unité (milliseconde, seconde,etc...) dépend de la base de temps de la fonction.

Variables de sortie		
Nom	Type	Commentaire
<i>OUT</i>	M	Sortie de la fonction
<i>TEMPO</i>	MW	Variable timer uniquement pour la fonction TON générique  Cette variable doit être déclarée dans l'application mère et définie comme "timer" cadencé selon la base choisie.

### 1.1.4. Exemple



Lorsque la condition *Cond* est validée depuis au moins 10 secondes, la sortie *Action* est activée, sinon elle est désactivée.





## 1.2. T\_BLINK (minute, secondes, millisecondes)

### 1.2.1. Compatibilité

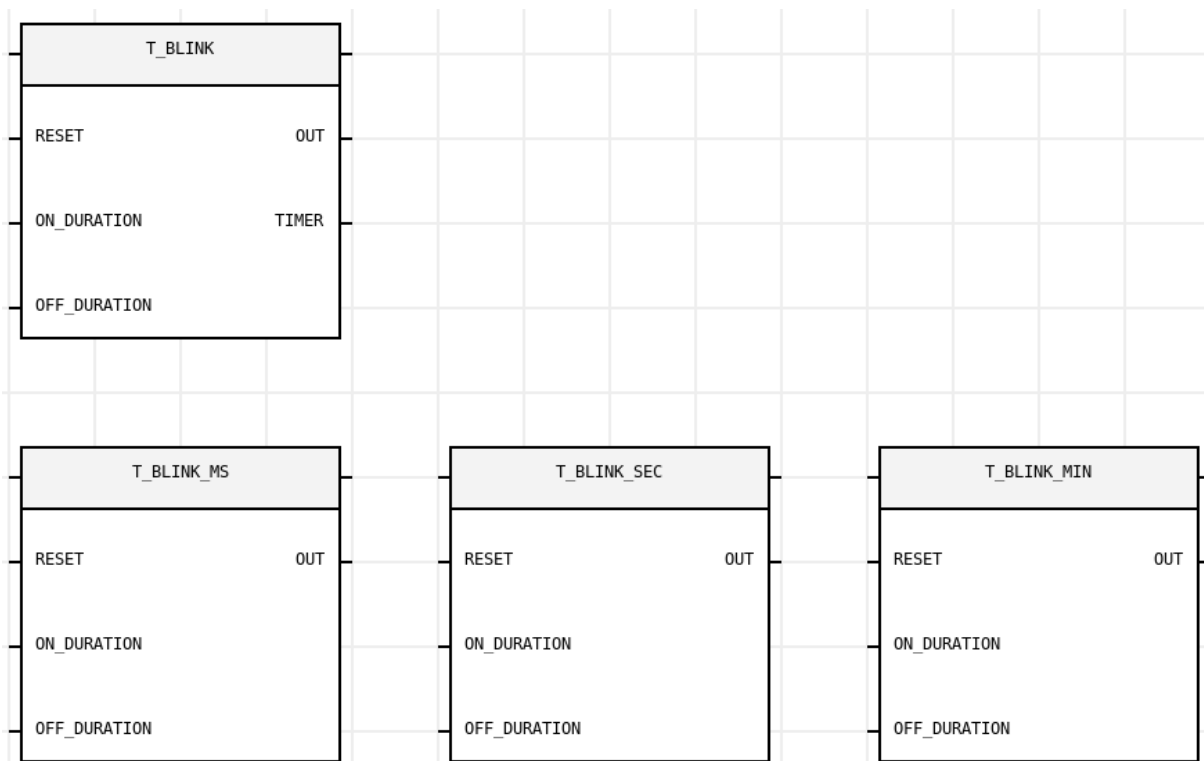
Toute version du code système

T\_BLINK générique : Toute version de µLadder

T\_BLINK base de temps : µLadder V7.9 ou plus

### 1.2.2. Description

Les tempos clignotantes permettent d'obtenir une sortie binaire qui change d'état périodiquement.

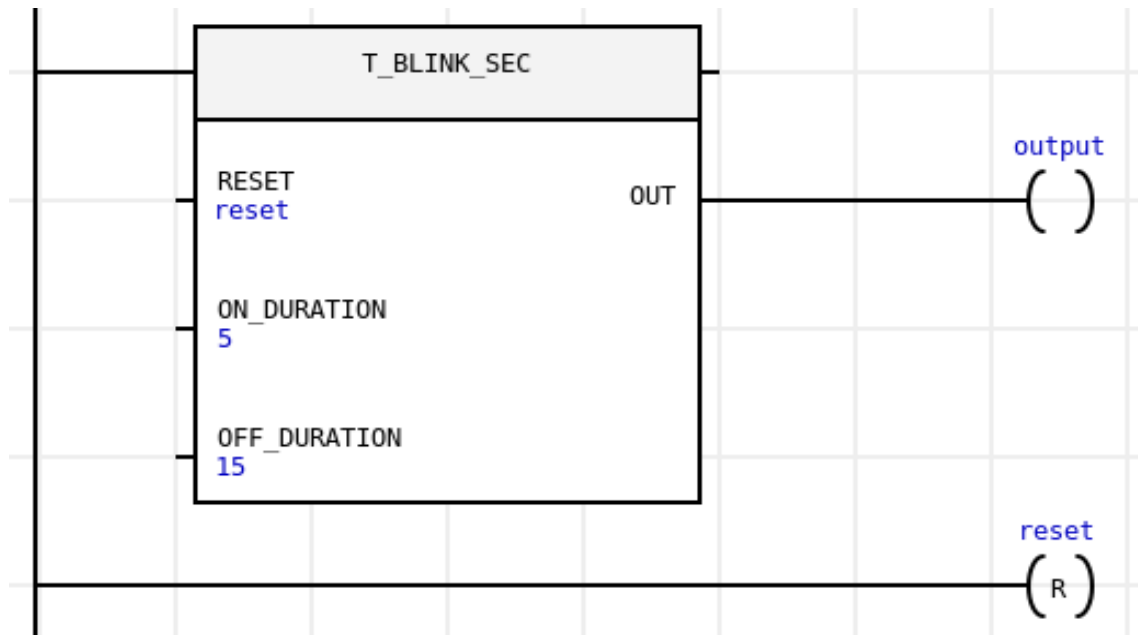


**1.2.3. Variables d'entrées/sorties**

Variables d'entrée		
Nom	Type	Commentaire
<i>RESET</i>	M	Un front montant sur cette entrée réinitialise la tempo et relance le cycle clignotant en début de phase OFF.  Entrée facultative.
<i>ON_DURATION</i>	MW	Durée de la phase ON
<i>OFF_DURATION</i>	MW	Durée de la phase OFF

Variables de sortie		
Nom	Type	Commentaire
<i>OUT</i>	M	Sortie de la fonction
<i>TIMER</i>	MW	Variable timer uniquement pour la fonction T_CLIGNO générique  Cette variable doit être déclarée dans l'application mère et définie comme "timer" cadencé selon la base choisie.

### 1.2.4. Exemple



Le bit *output* clignote avec une période de 20 secondes à raison de 5 secondes ON / 15 secondes OFF.

Lorsque l'on passe à 1 le bit *reset*, le cycle est réinitialisé et recommence le comptage à 0.

Dans cet exemple, le bit *reset* est immédiatement remis à 0 pour autoriser à nouveau d'autres réinitialisations.

## 2. Math

### 2.1. Min et Max

#### 2.1.1. Compatibilité

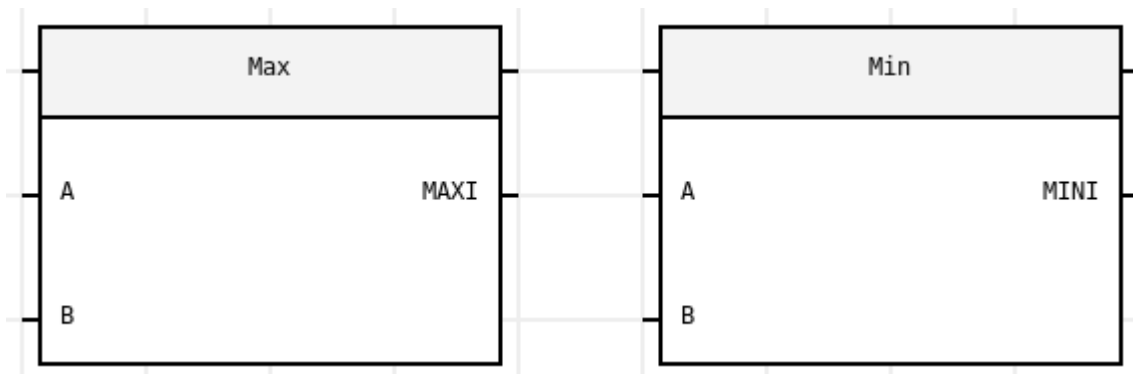
Toute version du code système

Toute version de µLadder

#### 2.1.2. Description

Renvoie la valeur mini ou maxi entre deux variables.

Fonctionne avec des entiers, longs ou flottants car les types sont convertis en entrée et en sortie.



### 2.1.3. Variables d'entrées/sorties

Variables d'entrée		
Nom	Type	Commentaire
<i>A</i>	MF	Variable A
<i>B</i>	MF	Variable B

Variables de sortie		
Nom	Type	Commentaire
<i>MINI / MAXI</i>	MF	Résultat du calcul

## 2.2. AbsDiff

### 2.2.1. Compatibilité

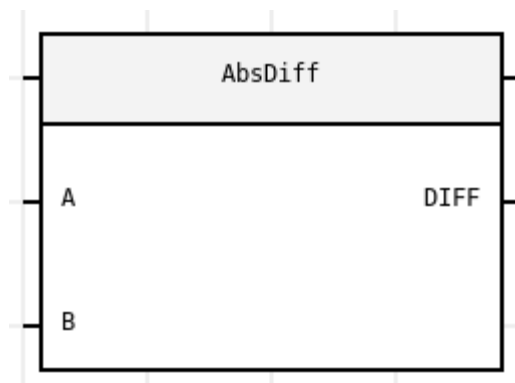
Toute version du code système

Toute version de µLadder

### 2.2.2. Description

Renvoie l'écart absolu entre deux valeurs.

Fonctionne avec des entiers, longs ou flottants car les types sont convertis en entrée et en sortie.



### 2.2.3. Variables d'entrées/sorties

Variables d'entrée		
Nom	Type	Commentaire
<i>A</i>	MF	Variable A
<i>B</i>	MF	Variable B

Variables de sortie		
Nom	Type	Commentaire
<i>DIFF</i>	MF	Résultat du calcul

### 3. Scales

#### 3.1. AIScale

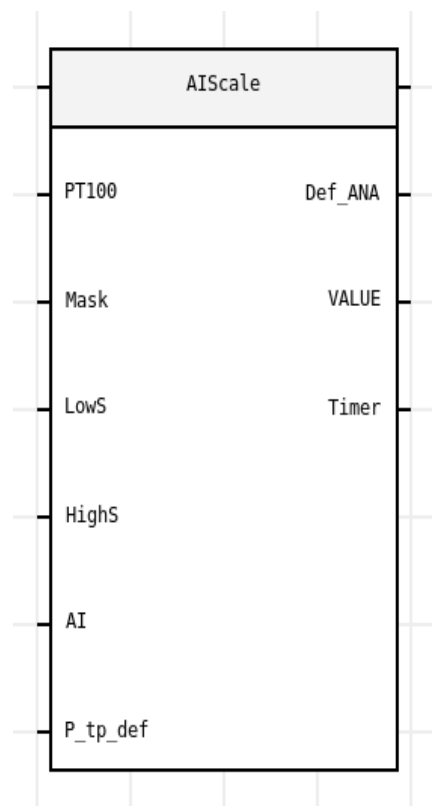
##### 3.1.1. Compatibilité

Toute version du code système

Toute version de µLadder

##### 3.1.2. Description

Cette fonction permet de mettre à l'échelle une entrée analogique de type PT100 ou 4-20mA.





3.1.3. Variables d'entrées/sorties

Variables d'entrée		
Nom	Type	Commentaire
<i>4_20mA</i>	M	1 si l'entrée est de type 4-20mA. 0 sinon.
<i>PT100</i>	M	1 si l'entrée est de type PT100. 0 sinon.
<i>Mask</i>	M	Bit à passer à 1 pour simuler un défaut
<i>LowS</i>	MF	Échelle basse pour la conversion 4-20 mA. Valeur correspondant à 4mA
<i>HighS</i>	MF	Échelle haute pour la conversion 4-20 mA. Valeur correspondant à 20mA
<i>AI</i>	MW	Valeur d'entrée analogique à mettre à l'échelle
<i>P_tp_def</i>	MW	Durée de la temporisation de défaut. Dans la même unité que la tempo (milliseconde, seconde ou minute)

Variables de sortie		
Nom	Type	Commentaire
<i>Def_ANA</i>	M	Défaut entrée analogique. Valeur hors plage sur une durée supérieur à la temporisation de défaut.
<i>VALUE</i>	MF	Mesure mise à l'échelle
<i>Timer</i>	MW	Variable timer pour gestion interne des tempos. Cette variable doit être déclarée dans l'application mère et définie comme "timer" cadencé selon la base choisie.

## 3.2. LinearScale

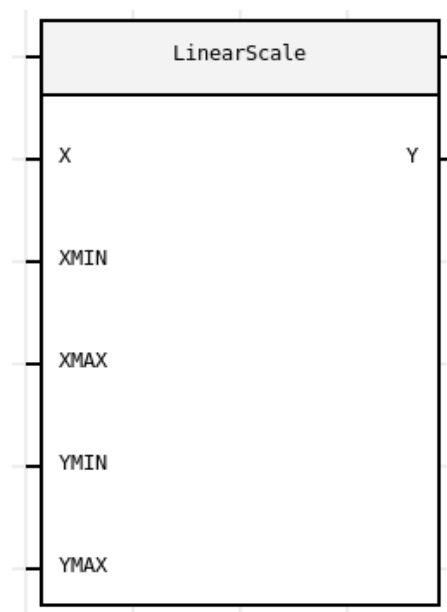
### 3.2.1. Compatibilité

Toute version du code système

Toute version de µLadder

### 3.2.2. Description

Cette fonction effectue une mise à l'échelle linéaire de type "Y=aX+b".



### 3.2.3. Variables d'entrées/sorties

Variables d'entrée		
Nom	Type	Commentaire
$X$	MF	Valeur X à mettre à l'échelle
$XMIN$	MF	Échelle de la valeur X
$XMAX$	MF	
$YMIN$	MF	Échelle de la valeur Y
$YMAX$	MF	

Variables de sortie		
Nom	Type	Commentaire
$Y$	MF	Valeur Y mise à l'échelle

## 4. Changements de valeur

### 4.1. ChangeDetect\_MW

#### 4.1.1. Compatibilité

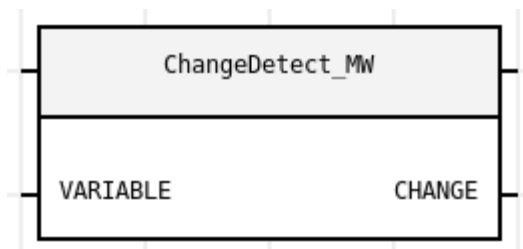
Toute version du code système

µLadder V7.9 ou plus

#### 4.1.2. Description

Permet de détecter le changement de valeur d'un MW d'un cycle à un autre.

C'est l'équivalent des fronts montant / descendant sur les bits.



#### 4.1.3. Variables d'entrées/sorties

Variables d'entrée		
Nom	Type	Commentaire
<i>VARIABLE</i>	MW	Variable à surveiller

<b>Variables de sortie</b>		
<b>Nom</b>	<b>Type</b>	<b>Commentaire</b>
<i>CHANGE</i>	M	Vaut 1 pendant le cycle ou la variable a changé de valeur. Vaut 0 sinon.

## 5. FILTERS AND AVERAGING

### 5.1. Moyenne glissante

#### 5.1.1. Compatibilité

Toute version du code système

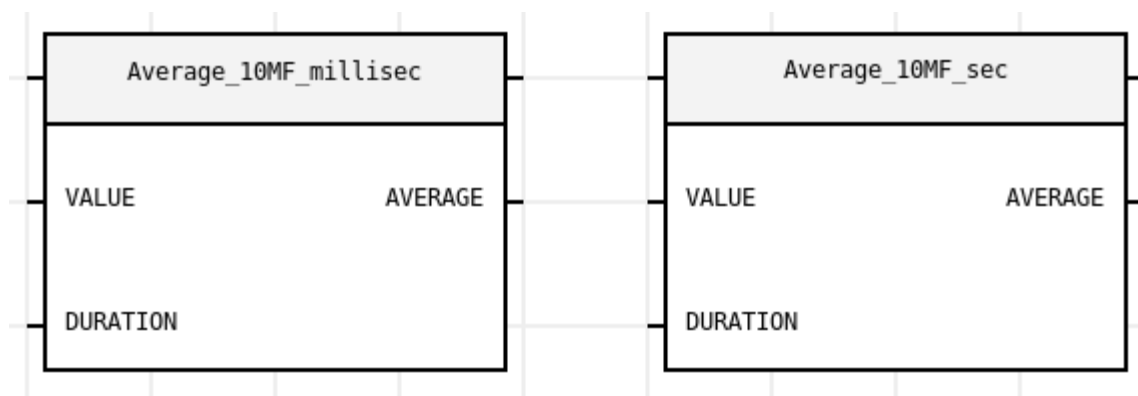
µLadder V7.9 ou plus

#### 5.1.2. Description

Les fonctions moyenne glissante calculent une valeur moyenne sur les X dernières secondes écoulées.

Elles fonctionnent par échantillonnage et sont donc limitée à un tableau de longueur fixe. Plus la durée de la période est élevée, moins le calcul est précis. Plus le tableau d'échantillonnage est grand, plus le calcul est précis mais plus le le bloc fonction est gourmand en mémoire RAM.

Les exemples ci-dessous fonctionnent sur 10 échantillons flottants (10MF) avec une base de temps à la seconde ou à la milliseconde.



### 5.1.3. Variables d'entrées/sorties

Variables d'entrée		
Nom	Type	Commentaire
<i>VALUE</i>	MF	Variable à filtrer
<i>DURATION</i>	MF	Durée de la période de calcul.  Cette valeur est exprimée en seconde si base de temps en secondes, en millisecondes si base temps en millisecondes, etc...

Variables de sortie		
Nom	Type	Commentaire
<i>AVERAGE</i>	MF	Valeur moyenne sur la période glissante.

## 6. Stations météo

### 6.1. WeatherStation

#### 6.1.1. Compatibilité

##### **WeatherStation\_V1.1.lad :**

Code système V3.2

µLadder V7.9 ou plus

##### **WeatherStation\_V1.2.lad :**

Code système V4.0 ou plus

µLadder V7.9 ou plus

#### 6.1.2. Description

Cette fonction communique avec une station météo SIREA connectée en Ethernet.

La socket doit être configuré en mode TCP.

Pour communiquer avec plusieurs stations météo, utiliser plusieurs blocs fonction avec des sockets différentes.



WeatherStation	
ip1	ready
ip2	windspeed
ip3	winddir
ip4	windgust
sock	pressure
	systemp
	temperature
	baraltitude
	windchill
	relhumidity
	abshumidity
	dewpoint

### 6.1.3. Variables d'entrées/sorties

Variables d'entrée		
Nom	Type	Commentaire
<i>ip1</i>	MW	Adresse IP de la station météo.
<i>ip2</i>	MW	Par exemple pour l'adresse ip 192.168.0.10 :
<i>ip3</i>	MW	ip1=192
<i>ip4</i>	MW	ip2=168
		ip3=0
		ip4=10
<i>sock</i>	MW	Numéro de la socket de l'automate.  A choisir parmi les différents sockets disponibles 0 à 3

Variables de sortie		
Nom	Type	Commentaire
<i>ready</i>	M	<p>État de la communication.</p> <p>0 = En cours</p> <p>1 = Terminé avec succès</p> <p>Une fois la communication terminée, elle recommence automatiquement dès le prochaine appel de la fonction soit au prochain cycle automate si la fonction est appelée à tout les cycle. La valeur 1 est donc active uniquement le temps d'un cycle.</p>
<i>windspeed</i>	MF	Données lues sur la station.
...	...	
<i>dewpoint</i>	MF	

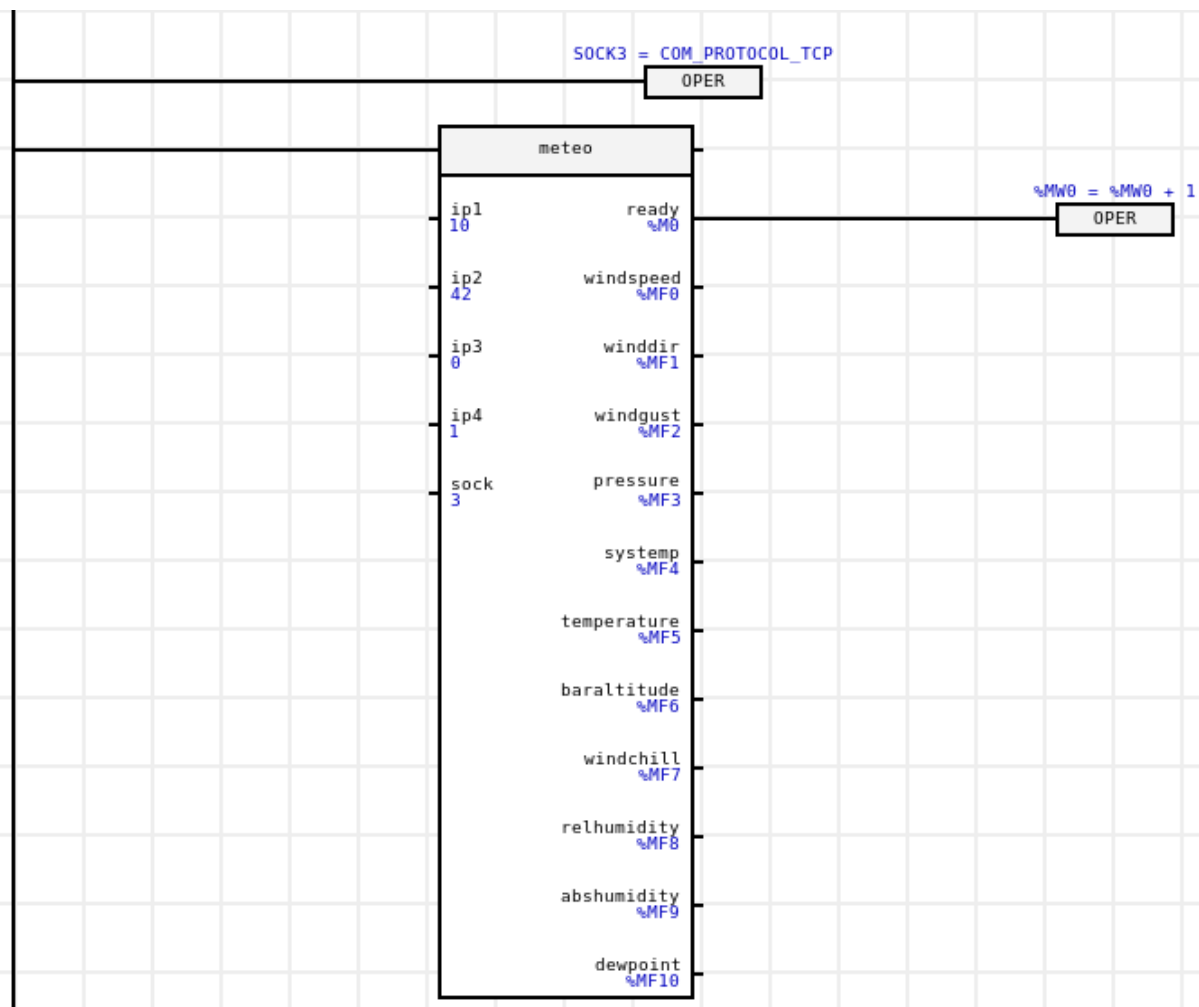
### 6.1.4. Exemple

La page et la fonction sont appelées inconditionnellement (chaque cycles automate).

On utilise dans cette exemple la socket n°3 de l'automate qui a bien été préalablement configurée en mode TCP.

%MW0 représente ici un compteur qui s'incrémente à chaque fois qu'un échange avec la station météo se termine avec succès. Si ce compteur ne bouge pas, c'est qu'il y a un problème de communication.

Les données mesurées par la station sont renvoyées dans les variables %MF0 à %MF10.



## 7. LCD DISPLAY

### 7.1. WriteText

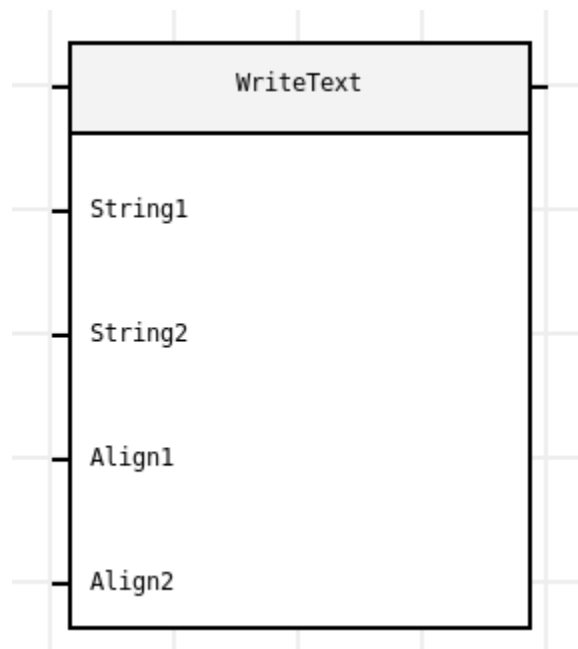
#### 7.1.1. Compatibilité

Code système V3.3 ou plus

µLadder V7.9 ou plus

#### 7.1.2. Description

Cette fonction affiche du texte sur les 2 lignes de l'écran LCD des µArm (Ecran présent sur les µArmA1, µArmA5 , µArmA7).



### 7.1.3. Variables d'entrées/sorties

Variables d'entrée		
Nom	Type	Commentaire
<i>String1</i>	MS	Chaîne de caractère à afficher sur la 1ère ligne
<i>String2</i>	MS	Chaîne de caractère à afficher sur la 2ème ligne
<i>Align1</i>	MW	Alignement horizontal de la 1ère ligne 0 = Gauche 1 = Centre 2 = Droite
<i>Align2</i>	MW	Alignement horizontal de la 2ème ligne 0 = Gauche 1 = Centre 2 = Droite

### 7.1.4. Exemple

Pour afficher l'état de l'entrée %I100.

